

Assignment1(day3)

Test-Driven Development (TDD) Process Infographic:-

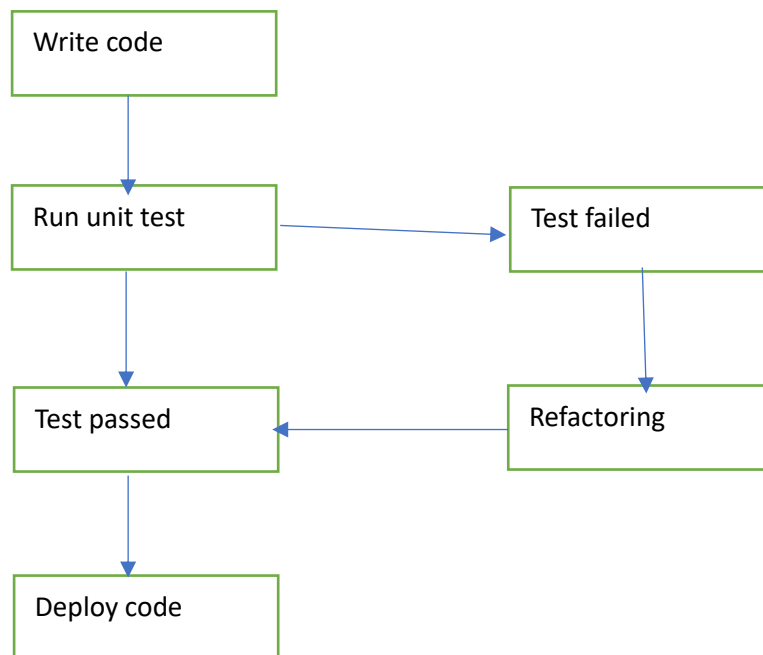
Write Test:- Before writing code, developers create automated tests based on specifications and requirements.

Run Test:- Execute the tests to confirm that they fail, as there's no code to satisfy them yet.

Write Code:- Implement the minimum code necessary to pass the test. This focuses on functionality.

Run Tests:- Execute tests again to validate the new code. If all tests pass, move to the next feature or refactor.

Refactor:- Improve code structure without changing its behavior. Tests ensure functionality remains intact.



Benefits of TDD:-

Bug Reduction:- Early detection and fixing of bugs through continuous testing.

Improved Code Quality:- Forces developers to write modular and maintainable code.

Reliability:- Confidence in software reliability due to comprehensive test coverage.

Faster Development:- Reduces debugging time and speeds up overall development process.

Why TDD Works:-

Feedback Loop:- Instant feedback on code changes ensures issues are caught early.

Design Guidance:- Encourages modular, loosely coupled designs which are easier to maintain.

Assignment2

Comparative Infographic: TDD vs. BDD vs. FDD

Test-Driven Development (TDD)

Approach:- Write tests before coding to define functionality.

Benefits:-

- Early bug detection.
- Improved code quality.
- Faster development with fewer defects.

Suitability:- Ideal for Agile environments and small-scale projects.

Behavior-Driven Development (BDD)

Approach:- Define behavior in plain language scenarios (Given-When-Then).

Benefits:-

- Improved collaboration between developers, QA, and stakeholders.
- Focus on user requirements.
- Clear documentation of features.

Suitability:- Best for complex projects with frequent changes and involving multiple stakeholders.

Feature-Driven Development (FDD)

Approach:- Develop features incrementally based on client priorities.

Benefits:-

- Structured development process.
- Emphasis on design and feature delivery.
- Scalable for large projects.

Suitability:- Suitable for large teams and projects requiring a structured approach to feature delivery.

Key Visual Elements:-

Icons:- Representing tests, behavior scenarios, and features.

Flow Diagrams:- Showing the iterative process for each methodology.

Graphs:- Comparing benefits like bug reduction, collaboration, and scalability.